N94-2 433

1993

NASA/ASEE SUMMER FACULTY FELLOWSHIP PROGRAM

MARSHALL SPACE FLIGHT CENTER
THE UNIVERSITY OF ALABAMA IN HUNTSVILLE

# NEURAL NETWORK-BASED CONTROL USING LYAPUNOV FUNCTIONS


| | |
|---|---|
| Prepared By: | Leon A. Luxemburg, Ph.D. |
| Academic Rank: | Assistant Professor Institution and |
| Department: | TAMU |
| | Marine Engineering |
| MSFC Colleague: | Henry B. Waites, Ph.D. |
| NASA/MSFC: | |
| Office: | Structures and Dynamics Laboratory |
| Division: | Control Systems |
| Branch: | Precision Pointing Systems |

# Introduction

Consider a linear nonminimal phase plant given as follows:

$$\dot{x} = Ax + Bu \tag{1}$$

$$y = Cx \tag{2}$$

The goals of this research effort are:

1. To develop an algorithm for offline stabilization of linear and nonlinear plants with known parameters by using a neural network controller.
2. The results of stabilization procedure must be rigorously tested mathematically.
3. The obtained controller should become linear controller which also stabilizes the plant when linearization of the neural network is performed.
4. Tracking of step inputs must be achieved.
5. Provide unified treatment of plant and controller dynamics in terms of differential equations rather than considering a hybrid discrete-continuous system.

To stabilize ( 1) we propose a neural network described by the following equations:

$$\dot{z} = g(z, u, y) \tag{3}$$

where the output of the net $o$ is given by $o = w_1^T y + w_2^T z$ and $u = o + ref$, where $ref$ is the reference input.

Definition of asymptotic stability of nonlinear system. Consider a plant-controller dynamical system given above in the phase space $R^n$ with state vector $(x^T, z^T)^T$. Then this controller stabilizes the plant with the region of stability $U$, $0 \in U \subset R^n$ if and only if disconnecting external input $ref$ results in convergence of any trajectory of combined plant-controller state space to 0.

The neural network consists of three layers: input layer, inner layer and the output layer with 5,4 and 2 nodes in these layers respectively. Sigmoid functions in the inner layers are chosen to be hyperbolic tangent functions $y(x) = (exp(x) - exp(-x))/(exp(x) + exp(-x))$. The layers are fully interconnected resulting in 28 weights. Additional weights are 4 weights for 2 two-dimensional vectors $w_1$, $w_2$ in the output $o$ above totalling 32 unknown weights. The $5 \times 4$ matrix of weights connecting input to inner layer is denoted by $E$ and the $4 \times 2$ matrix of weights connecting inner layer to the output layer is denoted by $D$. The total 32-dimensional weight vector is denoted by $r$.

To fully explain our approach we need to formulate two well known results about Lyapunov functions:

Result A: Let $\dot{x} = p(x)$, $x \in R^n$ be a differential equation on a bounded open set $U \subset R^n$ and let $p(0) = 0 \in U$. Let $h(x)$ be a continuous function on $U$ such that $h(x) > 0$ on $U$ and $h(0) = 0$. Let $< \nabla h(x), p(x) >< 0$ for all $x \in U$, where $\nabla h(x)$ denotes gradient of $h$ and $<,>$ denotes the scalar product in $R^n$. Then every trajectory of our differential equation with initial condition in $U$ converges to 0 as $t \to \infty$.

Result B: All the eigenvalues of matrix $T$ have negative real parts if and only if for any given positive definite symmetric matrix $N$ the matrix equation $T^T M + MT = -N$ has a unique positive definite symmetric solution $M$.

The basic underlying idea of the solution of stabilization problem using neural network controller is as follows: find a $6 \times 6$ matrix $M$ and the set of weights $r$ with dimension of $r$ being 32 such that $h(v) = v^T M v$ is the Lyapunov function in a neighborhood of 0 in a six dimensional state-space with the state vector $\binom{x}{z}$. This would require that the time derivative of $h$ , $\dot{h}(v) = v^T(T^T M + MT)v$ be a negative function on $U$ where $T$ is the Jacobian of the overall plant-controller dynamical system. Function $\dot{h}(x)$ depends altogether on 68 parameters: on vector $r$ and on vector $g$ which is such a vector that when arranged in a $6 \times 6$ matrix $G$ will satisfy the equation $GG^T = M$.

Our approach then is to start with random vector $r$ and random vector $g$ and form a gradient descent equation

$$\dot{q} = -\alpha \partial \dot{h} / \partial q \tag{4}$$

where $q$ is the six-dimensional state vector $q = \binom{x}{z}$, $\alpha$ is not a constant but a vector and in the formula above we consider the Hadamard product of $\alpha$ with the partial derivative of $\dot{h}$ by $q$. Also, $\alpha$ changes with time as the function $\dot{h}$ decreases.

While simulating the gradient descent equation we modify vectors $r$ and $g$ until function $\dot{h}$ above is negative on a neighborhood of 0.

To check that we have designed the stabilizing controller with the linear plant we need only to check that the egenvalues of the matrix $MT + T^T M$ are all negative. However, in this section we extend our method to nonlinear plants and show how to verify the stability in this case.

Algorithm for stabilization of nonlinear plants:

1. Stabilization of Jacobian at the equilibrium is done first and proceeds as in the case of linear plants. (Here we assume that the nonlinear plant has an equilibrium and we stabilize around this equilibrium).
2. After obtaining some open region of stability around the equilibrium as in part 1 we select points at random lying on concentric expanding spheres around this stable equilibrium and adjust the weights of neural net to achieve the negativity of the derivative of Lyapunov function. Lyapunov function $M$ is also given as a neural net.

Verification of stability of a given region for the given nonlinear plant and stabilizing neural net: Given the candidate for stability region $U$ and the Lyapunov function $h$ we can derive the upper bound $K$ on the partial derivatives of $\dot{h}$ with respect to state vector:

$$\partial\dot{h}/\partial w < K \tag{5}$$

where $w$ is the arbitrary point in $U$. If for every point $w \in U$ we have $\dot{h}(w) < -\beta$, $\beta > 0$ then, as follows from the Taylor's formula for multivariable functions, in the open ball of of radius $\beta/K$ the derivative $\dot{h}$ is negative. If we cover $U$ with the balls of radius $\beta/K$ then $\dot{h}$ is negative on $U$ insuring stability. This can also give us an estimate on the number of training points to achieve the stability.

Definition. Given a differential equation $\dot{x} = f(x)$, $x \in R^n$ a point $x_0$ is an equilibrium of order $k$, $k \leq n$ if $f(x_0) = 0$ and the Jacobian $\partial f(x_0)/\partial x$ at $x_0$ is nondegenerate and has exactly $k$ eigenvalues with positive real parts. By a stable manifold of $x_0$ we mean a union of all trajectories converging to $x_0$ as $t \to \infty$.

Definition. Consider the dynamical system $\dot{w} = f(w)$ described by a neural network-plant differential equations and having the Lyapunov function $h$. Let $U$ be the maximal set such that $U$ is connected, contains the origin of the state-space, $h$ is positive on $U$ and $\dot{h}$ is negative on $U$. Then $U$ is called the maximal stability region.

Theorem. In the notations of previous two definions let $\dot{w} = f(w)$ be a differential equation describing plant-neural network dynamical system and let $U$ be the maximal stability region for Lyapunov function $h$. Then

1. If $U$ is bounded then on the boundary of $U$ there are equilibria of all orders $k$, $0 \leq k \leq n$.

2. Under generic assumptions the boundary of $U$ is the union of stable manifolds of equilibrium points lying on the boundary.

3. Every trajectory on the boundary of $U$ converges to an equilibrium point as $t \to \infty$. If $U$ is bounded the the same is true for $t \to -\infty$.

4. The point on the boundary where the minimum of $h$ is achieved is an equilibrium point of order 1.

# Conclusions

We have successfully demonstrated how the problem of stabilization of plants can be reduced to a problem of approximation of functions. Neural networks have been shown to have approximating and interpolating properties. This approach is good for linear and nonlinear plants. Software has been generated to demonstrate this approach.

Directions for further research:

1. Generate faster software to utilize parallel processing features.

2. Improve algorithms to increase success rate for ill-conditioned plants such as the one considered. The convergence is successful for a random linear plant all the time.

3. Generate efficient software for nonlinear plants stabilization and tracking.

4. Study regions of stability and phase portraits of plant-neural controller and gradient descent learning differential equations.

5. Develop techniques for pole placing of linearized version of plant-neural controller system and of shaping the stability region.

# Acknowledgements

The substantial contributions to this work by Dr. Henry Waites and help by Mark Whorton is acknowledged and appreciated.